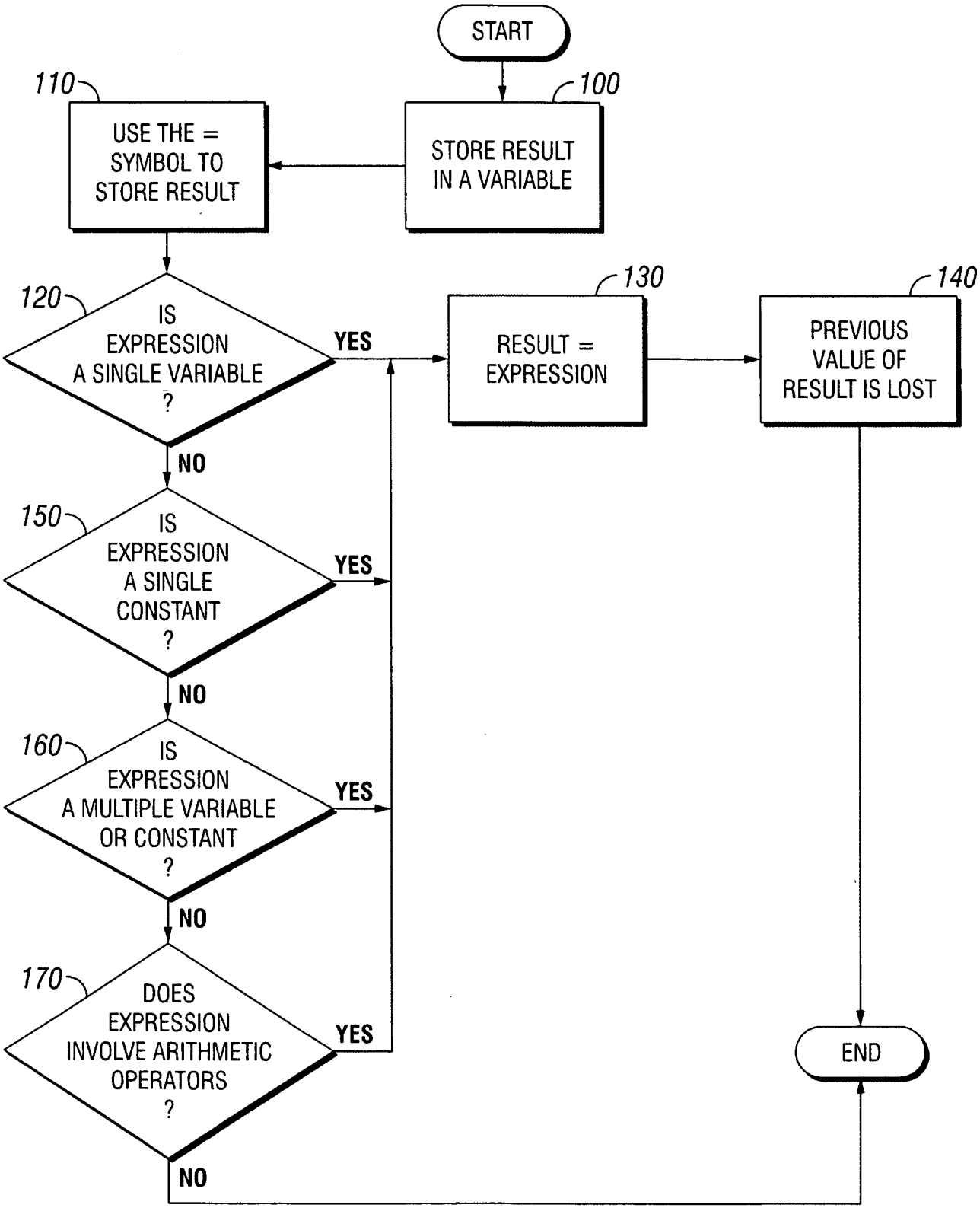
START
100
STORE RESULT IN A VARIABLE
110
USE THE = SYMBOL TO STORE RESULT
120
IS EXPRESSION A SINGLE VARIABLE ?
YES
NO
130
RESULT = EXPRESSION
140
PREVIOUS VALUE OF RESULT IS LOST
150
IS EXPRESSION A SINGLE CONSTANT ?
YES
NO
160
IS EXPRESSION A MULTIPLE VARIABLE OR CONSTANT ?
YES
NO
170
DOES EXPRESSION INVOLVE ARITHMETIC OPERATORS ?
YES
NO
END

FIG. 1

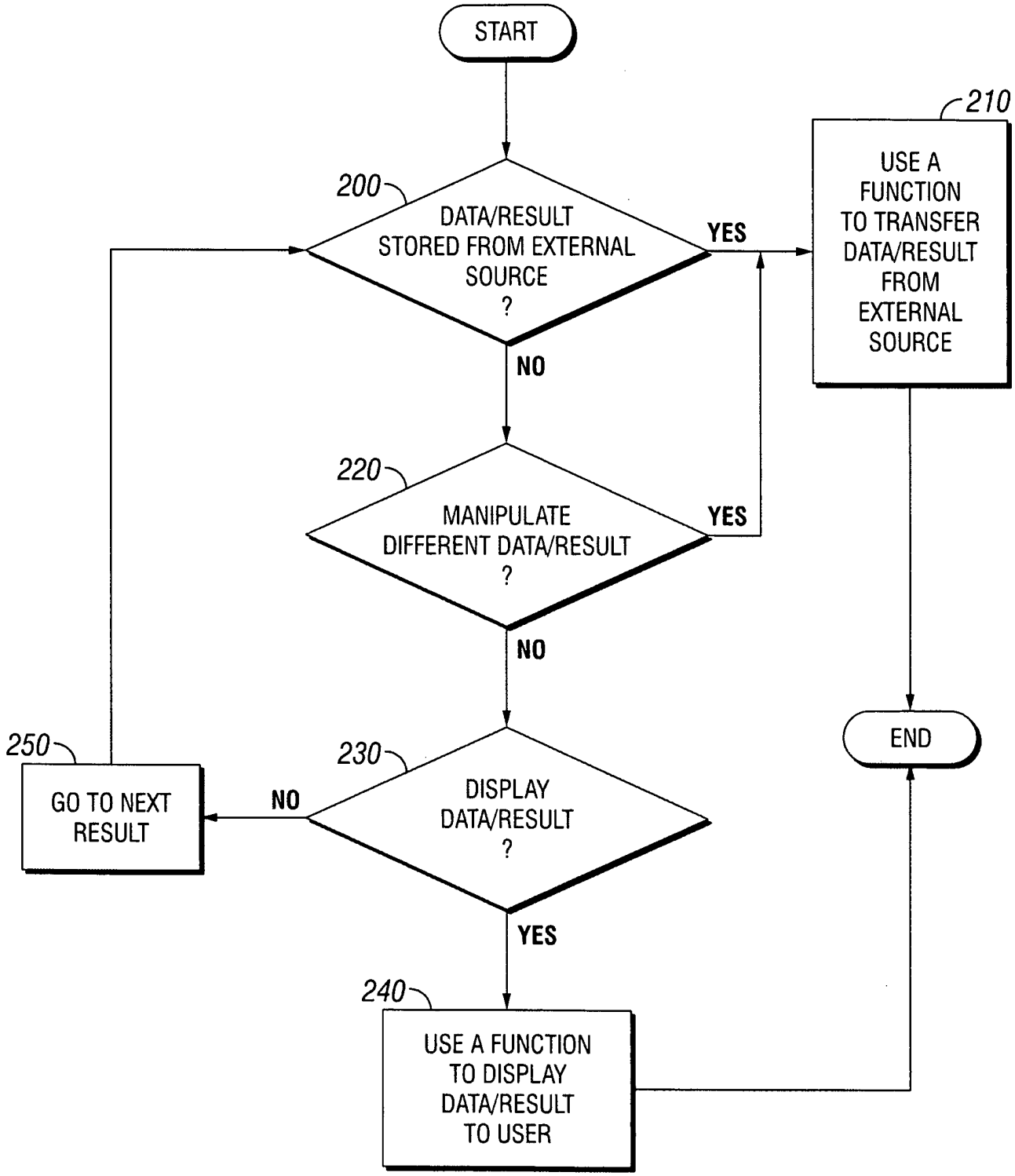
START
200
DATA/RESULT STORED FROM EXTERNAL SOURCE ?
YES
210
USE A FUNCTION TO TRANSFER DATA/RESULT FROM EXTERNAL SOURCE
NO
220
MANIPULATE DIFFERENT DATA/RESULT ?
YES
NO
END
250
GO TO NEXT RESULT
NO
230
DISPLAY DATA/RESULT ?
YES
240
USE A FUNCTION TO DISPLAY DATA/RESULT TO USER

FIG. 2

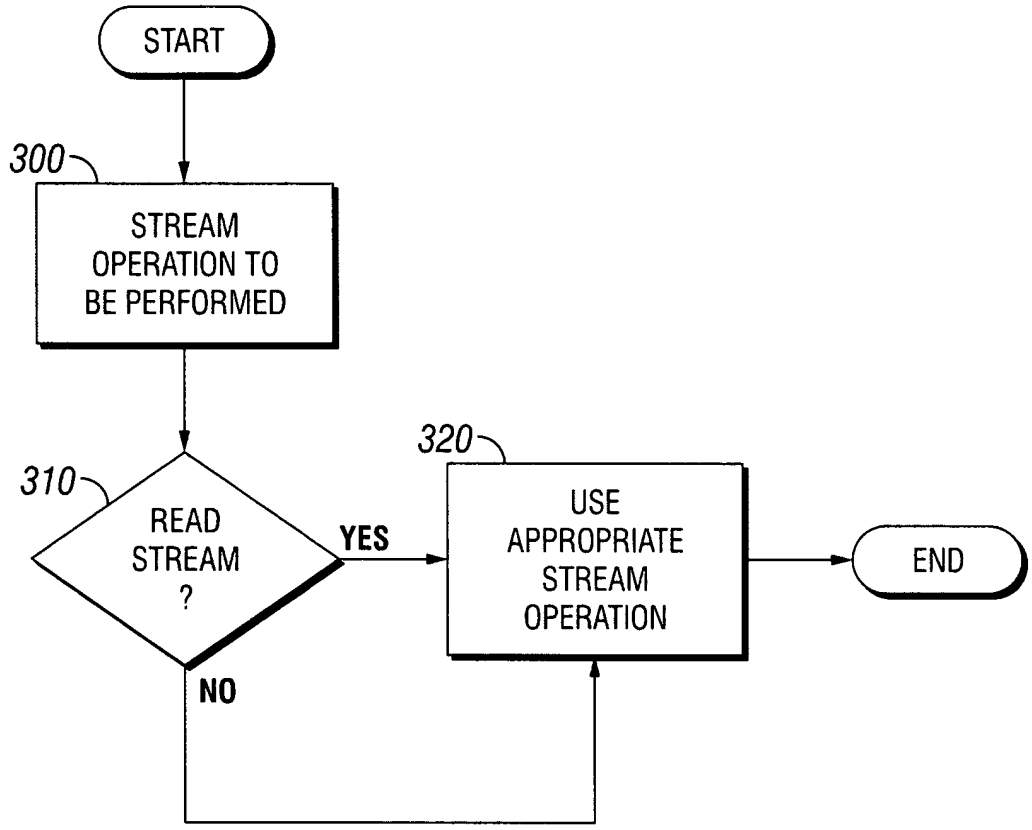
START
300
STREAM OPERATION TO BE PERFORMED
310
READ STREAM ?
YES
NO
320
USE APPROPRIATE STREAM OPERATION
END

FIG. 3

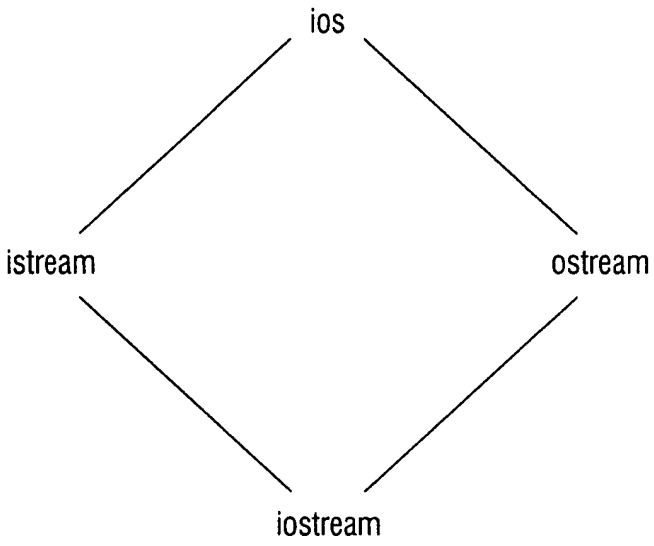
ios
istream
ostream
iostream

FIG. 4A

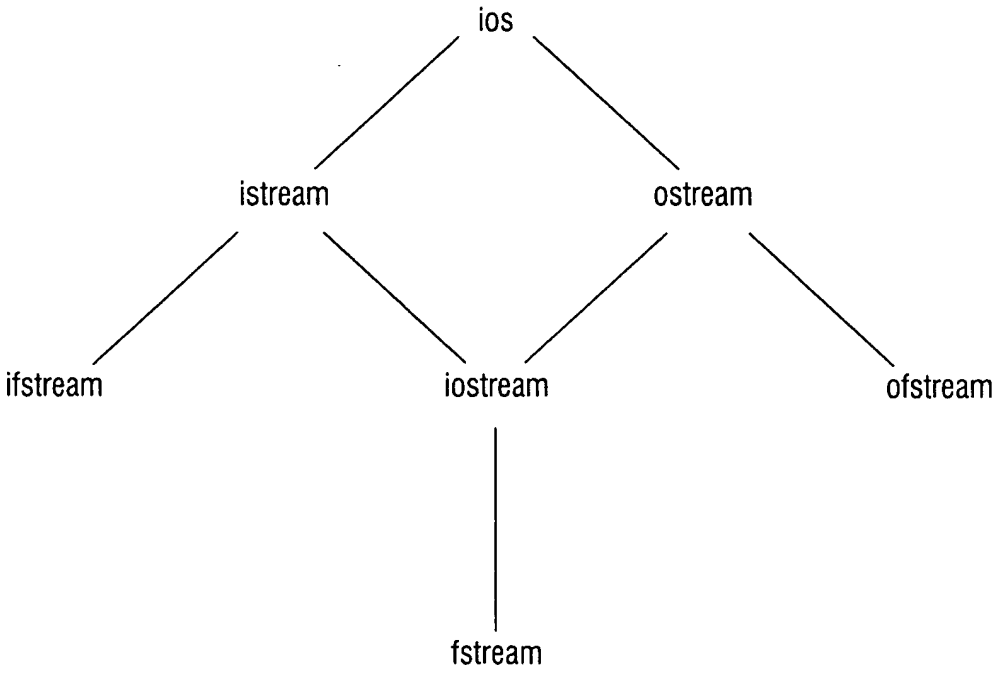
ios
istream
ostream
ifstream
iostream
ofstream
fstream

FIG. 4B

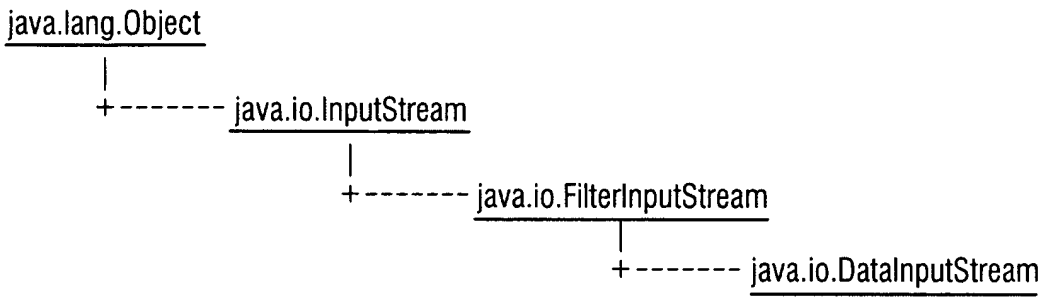
java.lang.Object
java.io.InputStream
java.io.FilterInputStream
java.io.DataInputStream

FIG. 5A

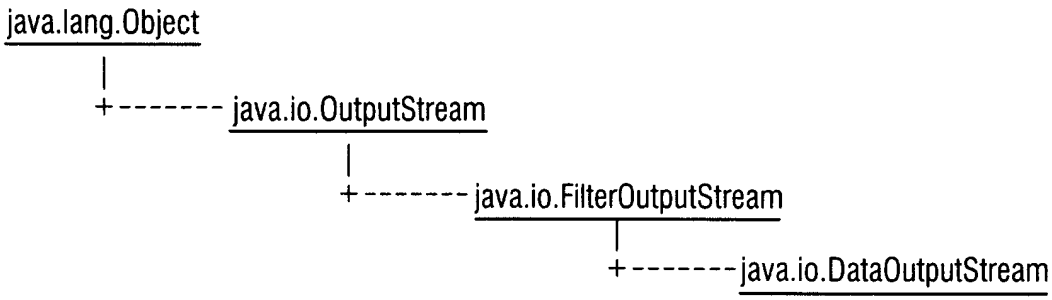
java.lang.Object
java.io.OutputStream
java.io.FilterOutputStream
java.io.DataOutputStream

FIG. 5B

Function	Purpose
openin (filename)	Open the named file for input
openout (filename)	Open the named file for output
openup (filename)	Open the names file for update(reading and writing)
openfd (fd)	Open the integer file descriptor as a stream
open (filename, mode)	Open the named file with the given integer mode

FIG. 6A

Function	Meaning
Network.open (addr,port)	Open an active network connection (TCP)
Network.openServer (addr,port, type)	Open a passive network connection (TCP or UDP)
Network.lookupName (name)	Consult a naming service to convert a network name to an address
Network.lookupAddress (ipaddr)	Consult a naming service to convert an integer IP address into a host name
Network.accept ()	Wait for and accept an incoming connection
Network.openSocket ()	Open a UDP client socket for sending datagrams
Network.send (socket, addr, port, buffer)	Send a UDP datagram to the given socket. The addr and port specify the recipient. The buffer is the string to send
Network.receive (socket, var addr, var port, maxbuffer = 4096)	Wait for an incoming UDP datagram. Blocks until a datagram is received, then returns the data as a string. Also sets the addr and port to the sending address. The maxbuffer argument specifies the max size of data that can be received
Network.peek (socket, var addr, var port, maxbuffer = 4096)	As receive, but don't extract the data from the network. A call to receiver() will read the data.
Network.formatIPAddress (addr)	Build a string of the form n.n.n.n out of an integer IP address.

FIG. 6B

Data type	Reading	Writing
integer	Decimal integer converted to binary	Decimal interger
string	Whole line read - terminated by line feed character which is discarded	Characters written to stream
real	ASCII for real value	Written as ASCII
char	Single character	Single character
vector	Each line of file appended to vector - line feed is retained	Each element written
map	cannot read	Elements written as first = second
enumeration constant	cannot read	Name of enumeration constant

FIG. 7

Operation	Result
close (stream)	The stream is closed.
select (stream, timeout)	Returns 1 if there is data waiting to be read from stream. Times out after timeout microseconds.
eof (stream)	Returns 1 if the stream is at the end of file.
flush (stream)	Flush the data remaining in the stream buffers.
getchar (stream)	Read a single character from the stream. Returns the character read.
getbuffer (stream)	Read all the available characters in the stream buffer. Returns a string containing all the characters.
availableChars (stream)	Returns the number of characters in the buffer.
setStreamAttribute (stream, attr, value)	Set the value of a stream attribute.
rewind (stream)	Rewind the stream to the start.
seek (stream, offset, whence)	Move to a new position in a seekable stream.

FIG. 8

Output type	Input type	Operation
integer	integer	copy left to right
	real	real converted to integer
	string	string converted to integer if possible, 0 otherwise
	vector	first element converted to integer
	map	first element converted to integer
	char	converted to integer
	block	integer set to address
	enumconst	index into enumeration
	object	call toInteger( ) if present, otherwise address of object
	stream	one integer read from stream
real	integer	converted to real
	real	copied
	string	string converted to real if possible, 0.0 otherwise
	vector	first element converted to real
	map	first element converted to real
	char	converted to integer then real
	block	converted to integer then real
	enumconst	converted to integer then real
	object	toReal( ) called if present, error otherwise
	stream	one floating point number read from stream
string	integer	converted to string
	real	converted to string
	string	copied
	vector	each element appended to string
	map	each element appended to string
	char	converted to string
	block	name of block
	enumconst	name of constant
	object	toString( ) called if present, blockname@address if not
	stream	one line read from stream
char	integer	truncated to 8 bits
	real	runtime error
	string	first character in string
	vector	first element converted to char
	map	first element converted to char
	block	first character of name
	enumconst	'A' = first const, 'B' = second, etc
	object	toChar( ) called if present, error if not
	stream	one char read from stream

FIG. 9A

Output type	Input type	Operation
vector	object	toVector( ) called if present, otherwise object appended to vector
	anything	appended to vector
map	object	toMap( ) called if present, otherwise appended
	anything	appended as {x =x}
function	scalar	function called with single argument
	vector	function called once for each element. Element passed as parameter
	map	Function called for each element. Function has to arguments for left and right of map pair.
	stream	function called for each line of input
thread		like function
class		like function only new object created for each
package		like class
enum		runtime error
enumconst		runtime error
object		runtime error
stream	integer	converted to decimal character sequence
	real	converted to floating point character sequence
	string	each character written
	char	single character written
	vector	each element written
	map	each element written as left = right
	block	block name written
	enumconst	name of constant written
	object	"object" + address written
	stream	stream copied

FIG. 9B

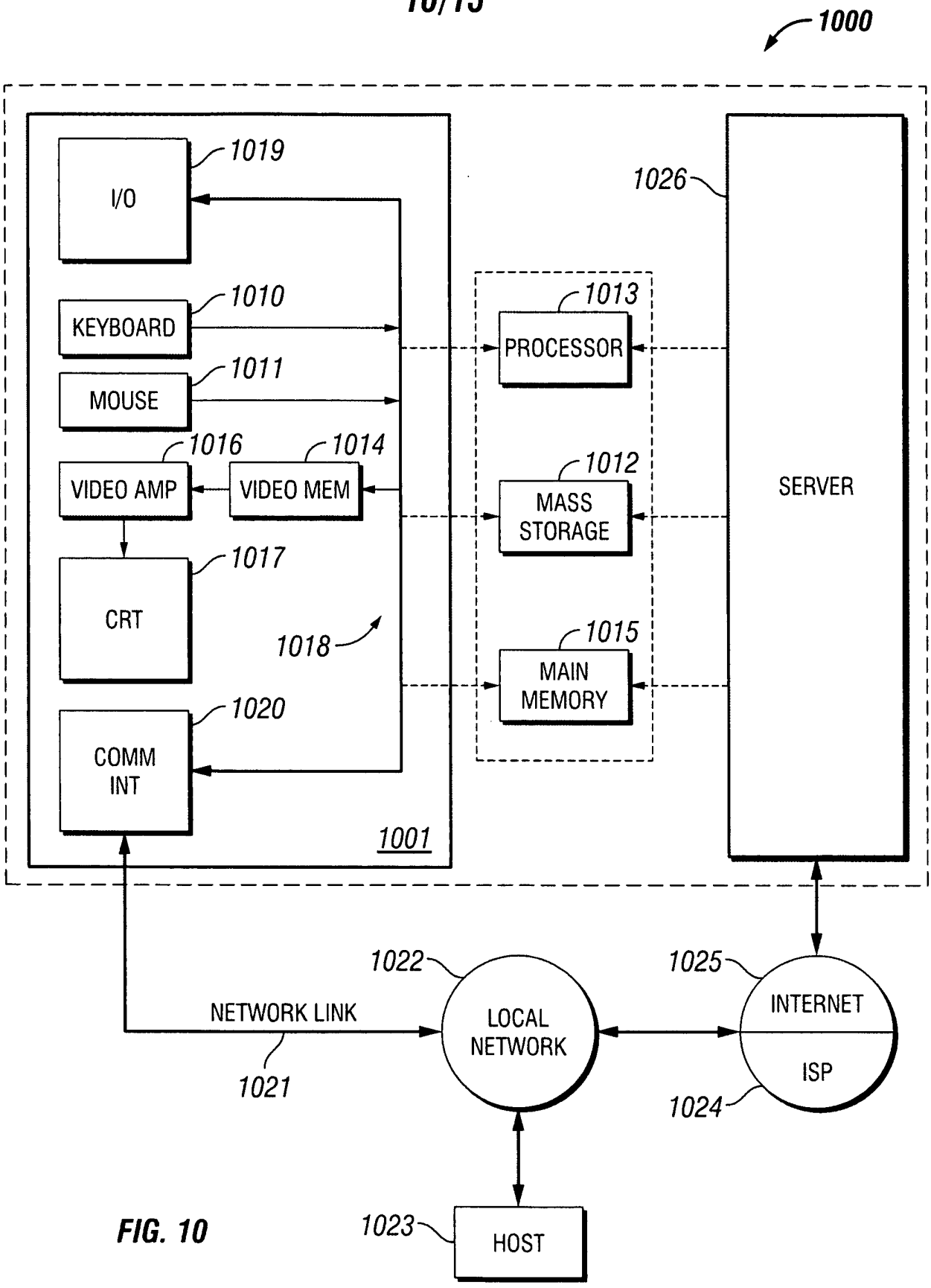
1000
1019
I/O
1026
1010
KEYBOARD
1013
PROCESSOR
1011
MOUSE
1016
1014
VIDEO AMP
VIDEO MEM
1012
MASS STORAGE
SERVER
1017
CRT
1018
1015
MAIN MEMORY
1020
COMM INT
1001
1022
1025
NETWORK LINK
LOCAL NETWORK
INTERNET
ISP
1021
1024
1023
HOST

FIG. 10

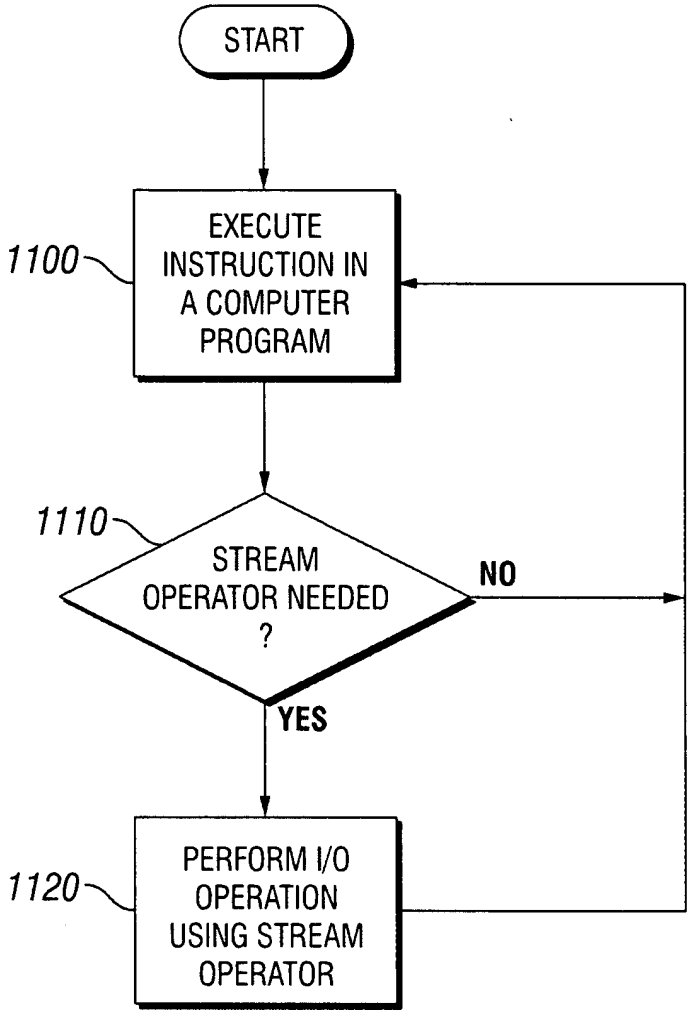
START
1100
EXECUTE INSTRUCTION IN A COMPUTER PROGRAM
1110
STREAM OPERATOR NEEDED ?
NO
YES
1120
PERFORM I/O OPERATION USING STREAM OPERATOR

FIG. 11

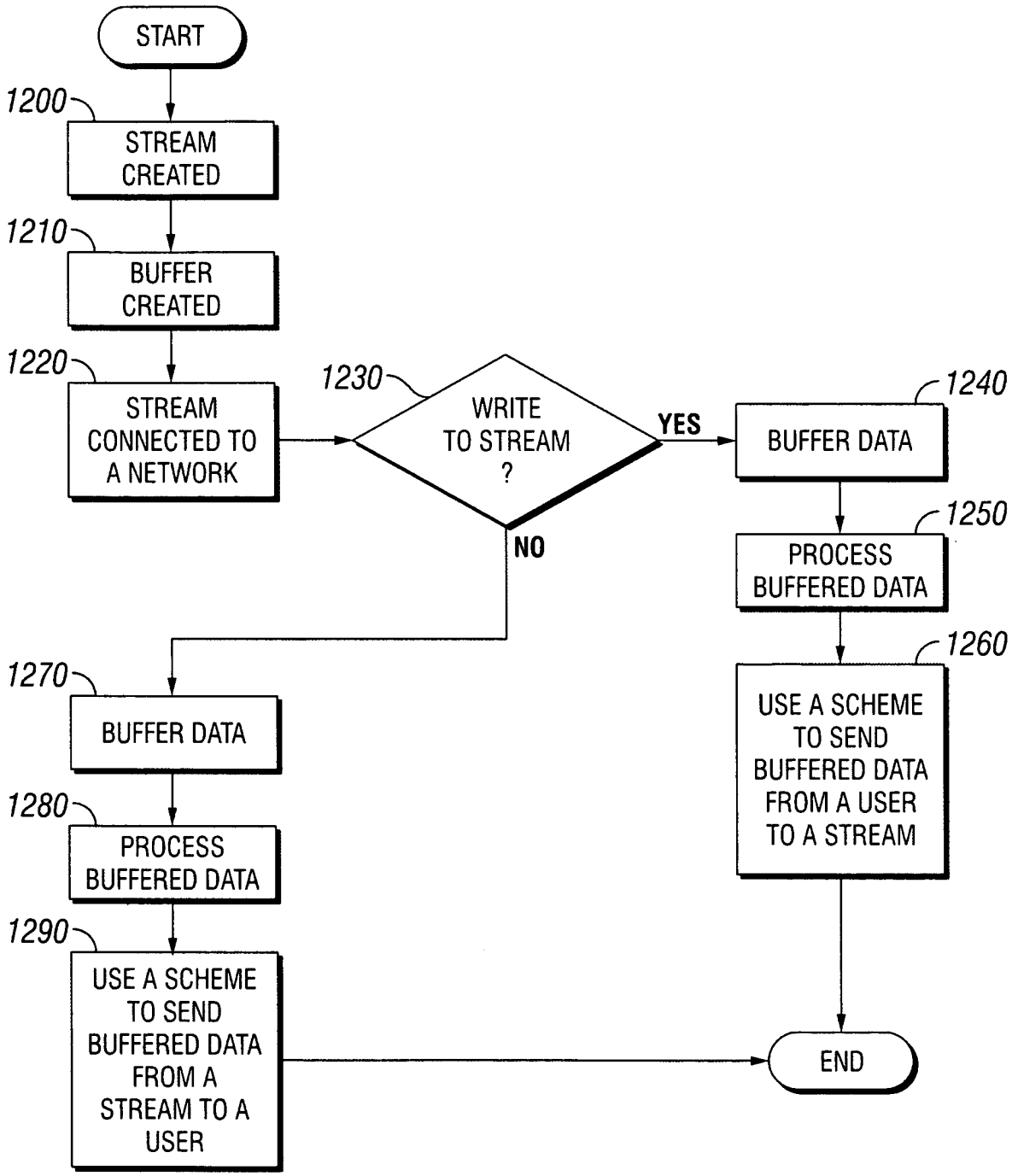
START
1200
STREAM CREATED
1210
BUFFER CREATED
1220
STREAM CONNECTED TO A NETWORK
1230
WRITE TO STREAM ?
YES
NO
1240
BUFFER DATA
1250
PROCESS BUFFERED DATA
1260
USE A SCHEME TO SEND BUFFERED DATA FROM A USER TO A STREAM
1270
BUFFER DATA
1280
PROCESS BUFFERED DATA
1290
USE A SCHEME TO SEND BUFFERED DATA FROM A STREAM TO A USER
END

FIG. 12

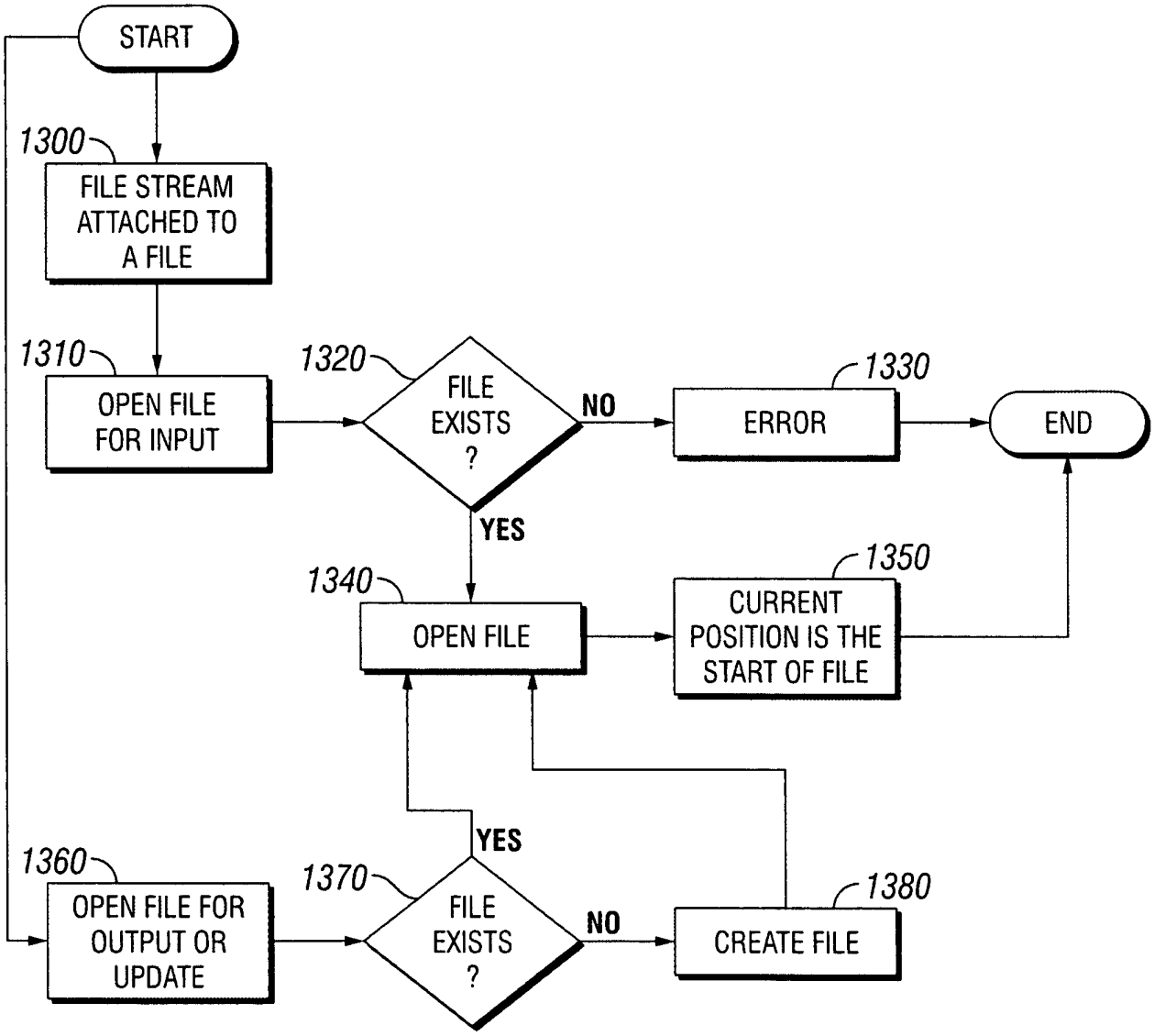
START
1300
FILE STREAM ATTACHED TO A FILE
1310
OPEN FILE FOR INPUT
1320
FILE EXISTS ?
NO
1330
ERROR
END
YES
1340
OPEN FILE
1350
CURRENT POSITION IS THE START OF FILE
YES
1360
OPEN FILE FOR OUTPUT OR UPDATE
1370
FILE EXISTS ?
NO
1380
CREATE FILE

FIG. 13

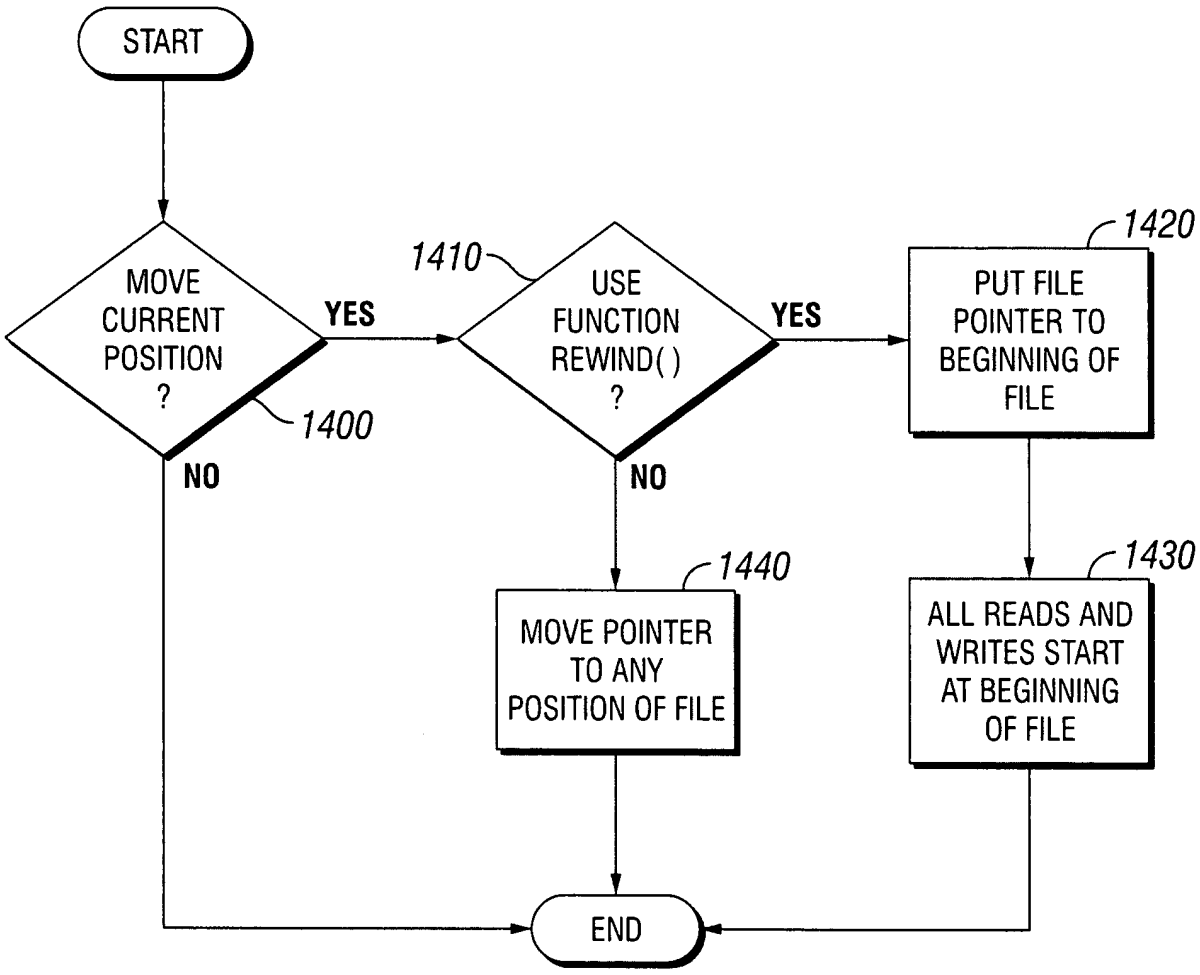
START
MOVE CURRENT POSITION ?
YES
NO
1400
1410
USE FUNCTION REWIND( ) ?
YES
NO
1420
PUT FILE POINTER TO BEGINNING OF FILE
1430
ALL READS AND WRITES START AT BEGINNING OF FILE
1440
MOVE POINTER TO ANY POSITION OF FILE
END

FIG. 14

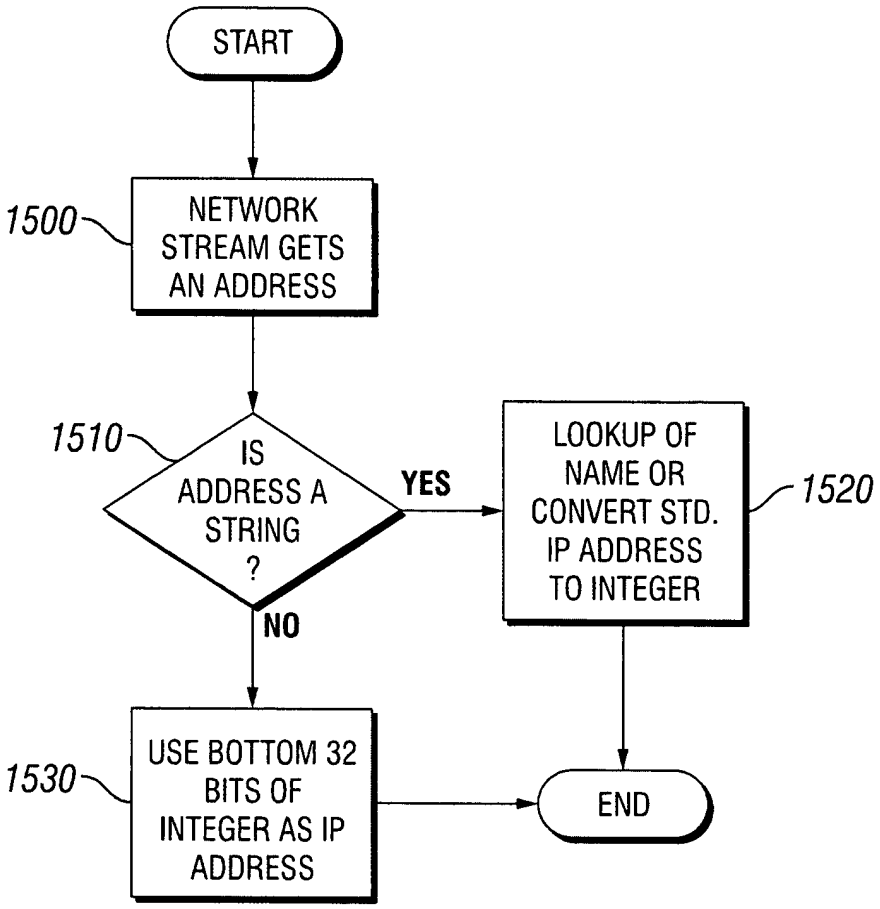
START
1500
NETWORK STREAM GETS AN ADDRESS
1510
IS ADDRESS A STRING ?
YES
NO
LOOKUP OF NAME OR CONVERT STD. IP ADDRESS TO INTEGER
1520
1530
USE BOTTOM 32 BITS OF INTEGER AS IP ADDRESS
END

FIG. 15